

FIG. 1

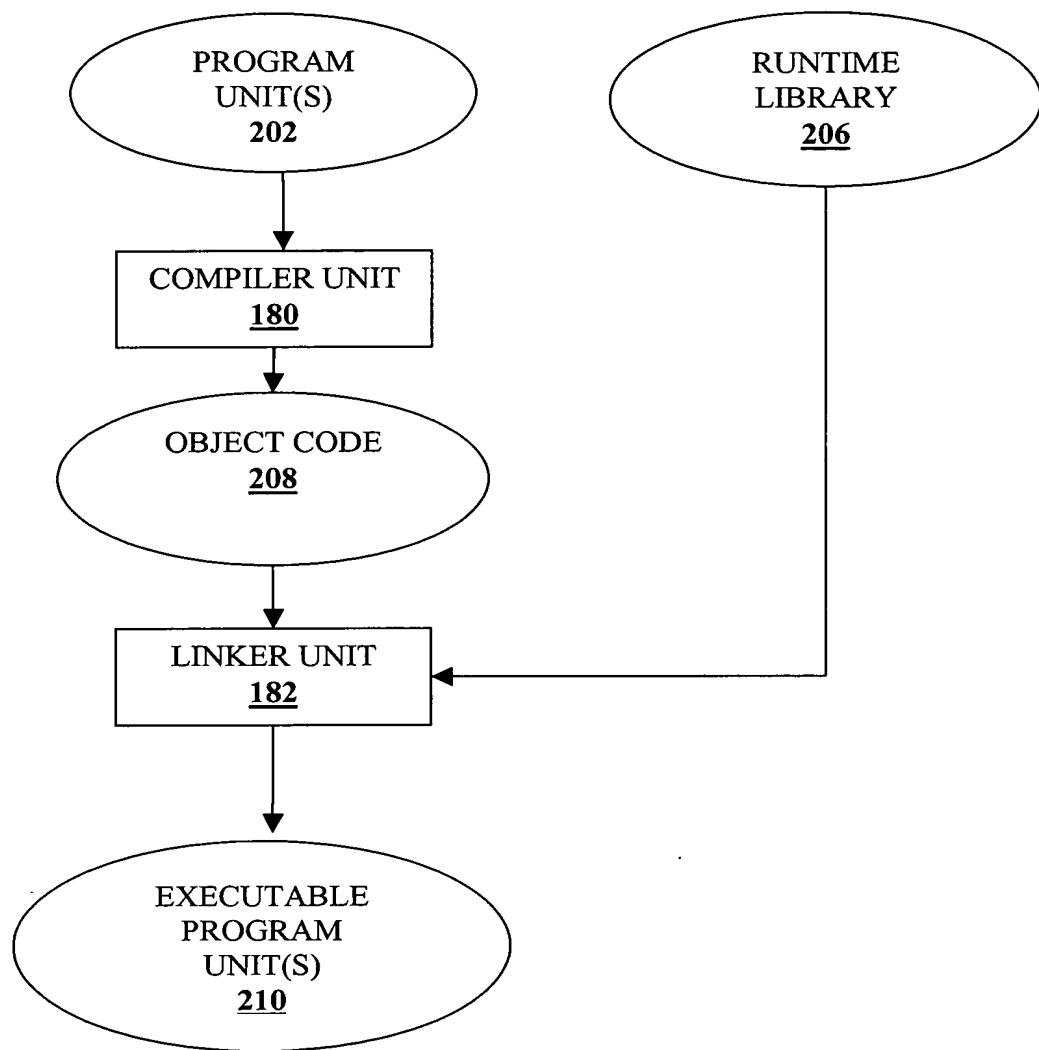


FIG. 2

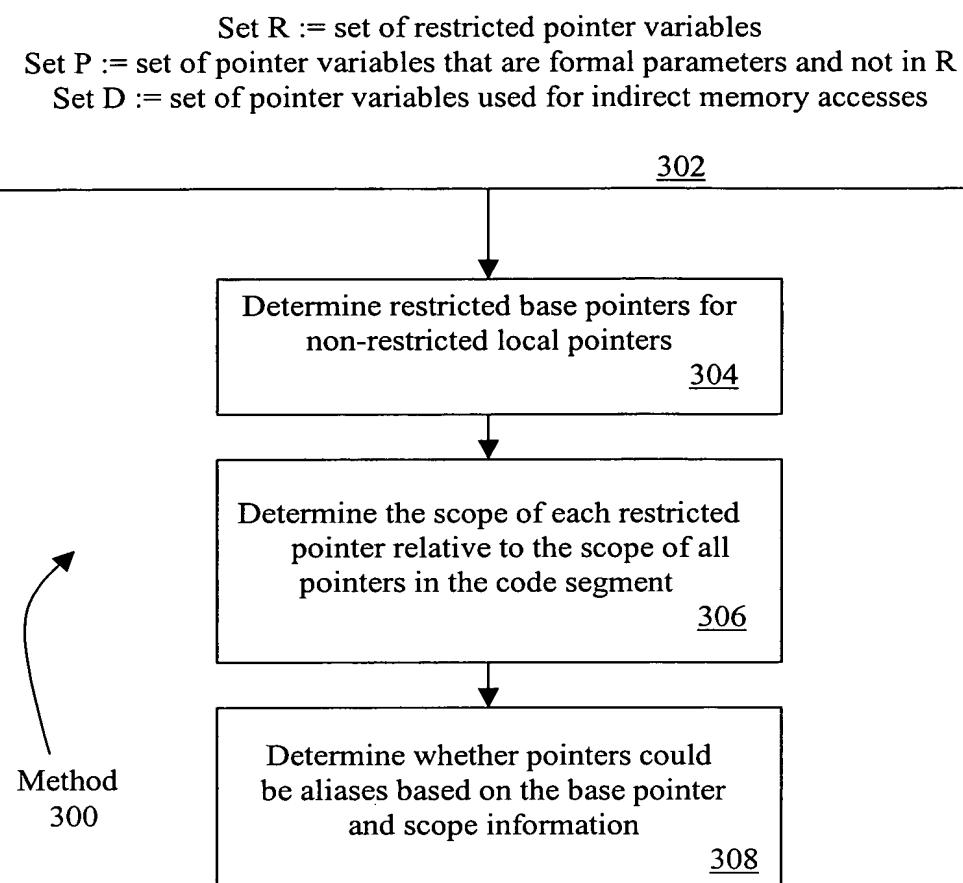


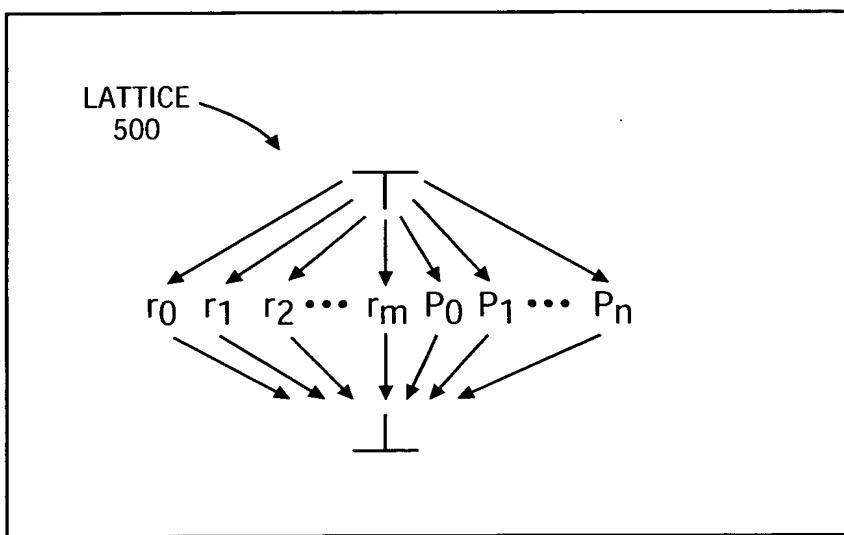
FIG. 3

Program 400

```
410 void bar( float * restrict a, float * x, int i, int j, int k ) {  
415   a[0] = x[0];  
420   {  
425     float * restrict b = a-k;  
430     float * restrict c = x+k;  
435     float * y = b+i;  
440     c[i] = *y;  
445   }  
450   {  
455     float * restrict d = a;  
460     {  
470       float * restrict e = x;  
475       d[j] = e[j];  
480     }  
490   }  
495 }
```

**FIG. 4**

042390.P11908



**FIG. 5**

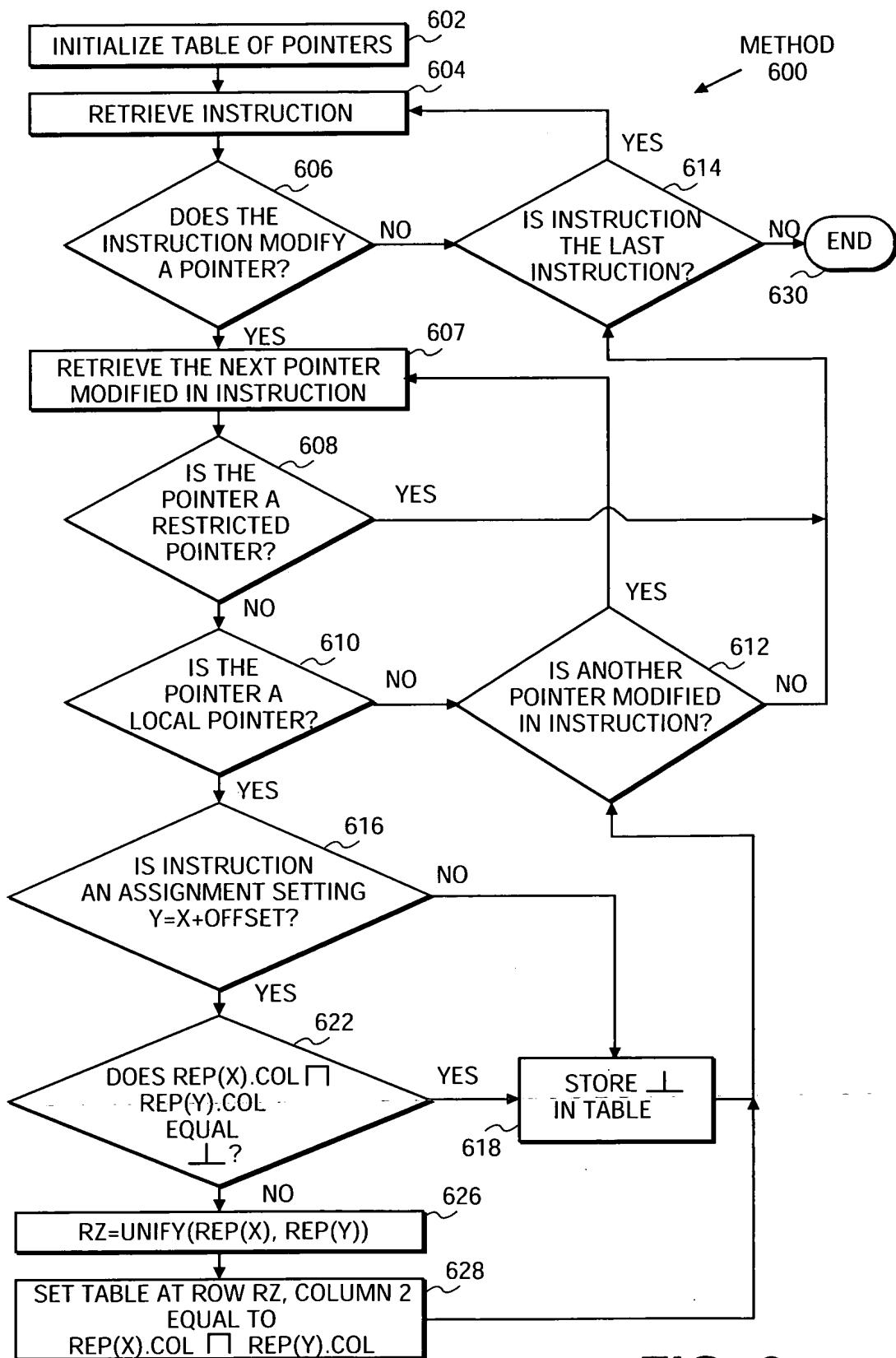


FIG. 6

procedure FLOW\_WALK

```
702 for each pointer variable  $w$  do
704   if  $w \in (R \cup P)$  then
706     REP( $w$ ).col =  $w$ ;
708   else
710     REP( $w$ ).col =  $T$ ;
712 enddo
```

702

```
714 for each instruction do
716   for each pointer variable  $y$  that might be modified by the instruction
718     if  $y$  is pointer variable that is restrict qualified then
720       //Ignore it.
722     else
724       if  $y$  is a local pointer variable then
726         if instruction is assignment that sets  $y$  to adjustment of  $x$  then
728           if  $REP(x).col \sqcap REP(y).col = \perp$  then
730             // Do not unify. Doing so just loses information.
732             REP( $y$ ).col =  $\perp$ 
734           else
736             //Target of  $y$  is same as target of  $x$ 
738             rz = UNIFY( $REP(y)$ ,  $REP(x)$ );
740             rz.col =  $REP(x).col \sqcap REP(y).col$ ;
742           endif
744         else
746           //Target of  $y$  is unknown
748             REP( $y$ ).col :=  $\perp$ ;
750           endif
752         endif
756       enddo
758     enddo

760 end FLOW_WALK
```

704

Pseudo Code 700

**FIG. 7**

Table 800

Pointer	REP(...).col
a	a
b	b
c	c
d	d
e	e
x	x
y	b

**FIG. 8**

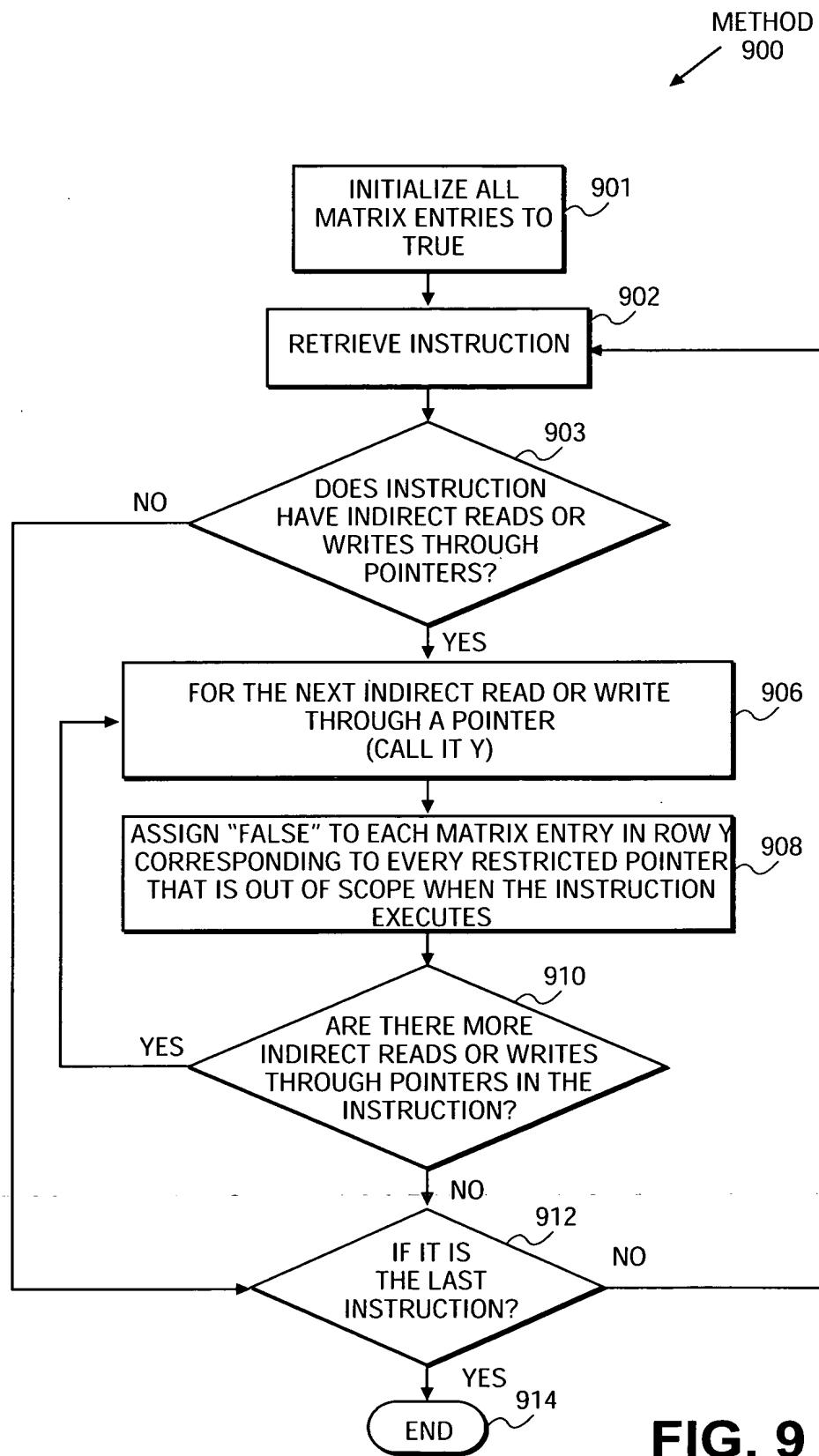


FIG. 9

procedure SCOPE\_WALK

```
1010 for each i in D do
1015   for each j in R do
1020     MATRIX[ROW(i),j] := true;
1025   enddo
1030 enddo
```

1002

```
1035 for each instruction x do
1040   for each indirect read or write through a pointer y do
1045     i := ROW(y);
1050     k := REP(y).col;
1055     if k ∈ (R ∪ P) then
1060       for each j in R do
1065         if j is not in scope when instruction x executes then
1070           MATRIX[i,j] := false;
1075         endif
1080       enddo
1090
1095     enddo
1096 endo
```

1004

end SCOPE\_WALK

Pseudo Code 1000

**FIG. 10**

RECORDED BY: ROBISON, ARCH D.

	a	b	c	d	e
a		x	x	x	x
b				x	x
c				x	x
d		x	x		
e		x	x		
x		x	x	x	x
y				x	x

Matrix 1100

**FIG. 11**

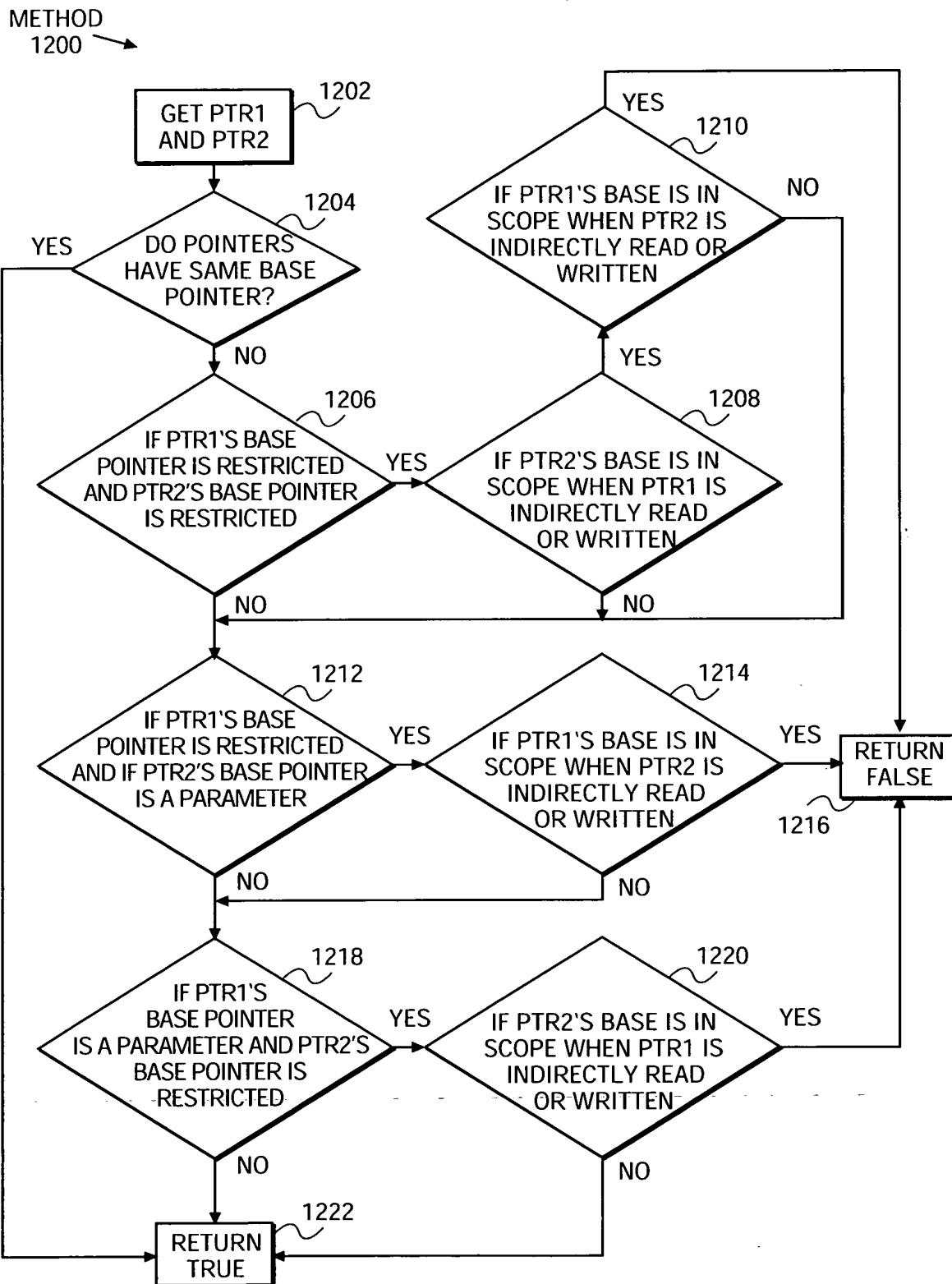


FIG. 12

Pseudo Code 1300

```
procedure COULD_TARGETS_ALIAS(x,y)
```

```
1320 i = REP(x).col;  
1302 j = REP(y).col;  
1306 if i=j then  
1308   return true;  
1310 endif
```

1302

```
1312 if i∈R and j∈R and MATRIX[ROW(x),j]=true  
1314 and MATRIX[ROW(y),i]=true then  
1316   return false;  
1318 endif
```

1304

```
1320 if i∈R and j∈P and MATRIX[ROW(y),i]=true then  
1322   return false;  
1324 endif
```

1306

```
1326 if j∈R and i∈P and MATRIX[ROW(x),j]=true then  
1328   returns false;  
1330 endif  
1340 return true;
```

1308

```
end COULD_TARGETS_ALIAS
```

FIG. 13